

Portable Program Review

Vol. 2, No. 3

March, 1986

FAST EDIT

In this article by Will Johnson, we learn the touch-typist's way to use Model 100 TEXT. -Ed

Most of us use our laptops -- at one time or another -- for notetaking or word processing. But have we learned some alternative keystrokes that could make text entry and editing faster and easier?

Learning these control codes is easy. Even if you have a high-powered word-processing program or ROM in your Model 100 or Tandy 200, you can improve your productivity, since a touch-typist won't want to move his or her hands from the keyboard to press an arrow key.

Vertical Motion

There are three basic sorts of vertical cursor movements -- full file, screen (either 8-or 16-line) and single line. These movements

are usually made with Ctrl, Shift or plain arrows... but here's another way:

file
top Ctrl-W Ctrl-Up
bottom Ctrl-Z Ctrl-Down

screen
top Ctrl-T Shift-Up
bottom Ctrl-B Shift-Down

line
up Ctrl-E Up
down Ctrl-X Down

Notice that W and Z form a vertical pair of letters, as do B and T, and E and X.

Horizontal Motion

We can also move the cursor left or right to the extremes, by word or one character at a time.

screen
left Ctrl-Q Ctrl-Left
right Ctrl-R Ctrl-Right

word
left Ctrl-A Shift-Left
right Ctrl-F Shift-Right

character
left Ctrl-S Left
right Ctrl-D Right

Again, look at the keyboard. Q and R are on the same horizontal alignment, as are A and F, and S and D.

Remember that Ctrl-L can insert a linefeed, too!

If you can learn these codes,

IN THIS ISSUE:

FAST EDIT 1

THE PROGRAM KILLER..... 2

LEARNING MADE EASY 5

and if you're a good typist, then you should be able to work the Model 100 even faster -- because you won't be moving your fingers too far from the normal home position.

THE PROGRAM KILLER

This excellent utility by Andy Barbolla reminds me of my old Model III days, where we had a file-cleaning called PURGE. But this program is even better. -Ed

When I'm through working on a specific application on the Model 100 and have saved all the pertinent information, I'm left with a bunch of now useless programs or files staring at me from the main menu -- wasting precious bytes. I'd like to move on to my next project with all unnecessary files out of the way, but the thought of typing kill "filename.extension" twenty times is intimidating. I tried writing a sample program to help me, but killing a BASIC file from a program ends the program at that point. So a simple program just won't do it. I could cold start the computer but I'd lose all of my utilities.

Enter KILLR. It will clear out up to thirteen preselected files in about forty-five seconds and you don't even have to type in a single file name.

When you run KILLR, the screen will clear, RAM files will be listed across the screen and the message "PLEASE WAIT" will appear while KILLR reads the file names from the screen and puts the names into an array. After a few moments the names are redisplayed four across (instead of three across as before) with the addition of two commands (CLEAR and KILL) after

the last filename. The top left file will be rapidly flashing between reverse and normal characters. At this point the program somewhat resembles the main menu; the arrow keys and spacebar move the cursor from file to file.

Pressing the Enter key marks the file under the cursor for destruction. That file name changes to inverse display, to remind you that you've selected it. Continue around the screen, choosing files to be killed just by hitting the Enter key. Up to thirteen files may be deleted at one time. If you attempt to choose more than thirteen, the program will simply not oblige. If you have picked a filename by accident or have changed your mind, move the cursor over to the command CLEAR and press Enter. The program will re-run from scratch.

After you've chosen all the files that you wish to delete, move the cursor over to the command KILL and press Enter. The files you have selected are displayed and the program asks "KILL ABOVE?" At this point, double-check the files listed. If they're right press the Y key -- any other entry starts from scratch. At any time during the use of KILLR, the program may be aborted by pressing function key 1 -- this will bail you out if you pressed Y accidentally.

A PLEASE WAIT message will be displayed as the program once again reads the file names off of the screen and places them into an array. It clears the screen and reprints all of the elements of the array except the first, which it proceeds to KILL. The program ends and the phrase RUN40 is displayed, which restarts the program two lines back. The process continues over and over until all filenames are killed. This Program Mode/Command Mode

loop kills all of the selected files and then returns to the menu.

What Happened?

Backing up a bit: Attempting to kill a BASIC file from a BASIC program results in the program stopping at that point. So an easy one or two line program just won't allow multiple kills for BASIC files (although it will work for .DO and .CO files). After the program ends, the computer must be tricked into thinking that the restart command RUN40 was typed in by hand. This is accomplished by filling the keyboard buffer with the appropriate characters while still in the program mode and then ending the program. The contents of the keyboard buffer is dumped to the screen and executed in the Command Mode.

This powerful trick, sometimes called the Dynamic Keyboard method, has been used for years on a variety of different machines. It allows you to execute various commands and then re-take control of the computer even though the program has ended.

Fast Change

This program can be modified to do a great many things:

Adding a Rename capability to the KILLR program enhances its usefulness as a utility program. Although only one program can be renamed at a time, it's still a handy feature.

Changes and additions:

```
6 FG=1:GOSUB60:ONINSTR(CHR$(29)
+CHR$(28)+" "+CHR$(30)+CHR$
(31)+CHR$(13)+"Rr",I$):GOTO8,
10,10,12,14,18,60,60:GOTO6
```

```
60 O$=A$(T):PRINT@245,"RENAME "
O$" AS:";:INPUTN$:IFN$="" THEN
PRINT@245,"":GOTO6
```

```
ELSENAMEO$ASN$+RIGHT$(O$,3)
:RUN
```

Selection of a file to be renamed is the same as described earlier; just position the cursor over the desired name. Pressing the R key will prompt you with the old filename and ask for a new one. Hitting Enter without a name will abort, causing the prompt to disappear and the cursor to start flashing again. Typing in a name will then change the filename and re-run the program. DO NOT type in the extension -- it is already provided.

Another interesting modification will allow this program to autoload multiple files from the portable disk drive using the new POWR-DISK software by Acroatix Inc. This really makes the loading many files in from the disk a pleasure. The modification uses the same concept as KILLR except that a temporary DO file is created to read in the disk filenames. Disk names are read from the DO file and placed into an array. When all the names have been read into the array the temp file is no longer needed ~~and therefore killed~~. The disk names are printed to the screen and the program continues as described earlier.

Changes and additions:

```
4 CLEAR1500:DIMZ$(32):DIMA$(32)
:DIMK$(14):KEYON:ONKEYGOSUB36:
RO$=CHR$(27)+"p":RF$=CHR$(27)+
"q":W$=RO$+"PLEASE WAIT"+RF$:
CLS:GOSUB100:PRINT@294,W$;:
GOSUB44:J$="<LOAD>":L$=
"<CLEAR>":A$(I)=J$:A$(I+1)=L$:
I=I+2:CLS:FORZ=0TOI:PRINT@Z
*10,A$(Z);:NEXT:GOSUB32
```

```
38 CLS:PRINTRF$:FORZ=0TOK:PRINT@Z
*10,K$(Z):NEXT:PRINT@254,"LOAD
ABOVE?":GOSUB52:IFI$="Y"ORI$=
"y"THENPRINT@254,W$ELSERUN
```



```

42 CLS:FORZ=1TOI:PRINT@(Z-1)*10,
  A$(Z):NEXT:PRINT@80,"":E$=
  "RUN40"+CHR$(13):L=LEN(E$):FOR
  P=-85TO-85+(L*2)-1STEP2:E=E+1
  :POKEP+1,0:POKEP,ASC(MID$(E$,
  E,1)):NEXT:POKE-86,L:LOADM": "+
  A$(0),F:END

```

```
100 LFILES TO "TEMP"
```

```
102 OPEN"TEMP"FORINPUTAS1
```

```
104 IFEOF(1)THENCLOSE:GOTO108ELSE
  Z$(V)=INPUT$(9,1):V=V+1

```

```
106 GOSUB120:IFI$=CHR$(13)THEN
  GOSUB120:GOTO104ELSE106

```

```
108 FORZ=0TOV:PRINT@Z*10,Z$(Z);:
  NEXT

```

```
110 KILL"TEMP.DO":RETURN
```

```
120 I$=INPUT$(1,1):RETURN
```

There are many other possibilities for the enhancement of KILLR. It can be modified to become a full-featured RAM file utility, a disk operating system or who knows? See Table 1 for a line by line breakdown -- and start experimenting!

```

4   Initialize, print files,
    read files into array A$(I),
    print elements four across
    on screen
6   Cursor key input and set
    flash flag
8-16 Move cursor
18-20 KILL or CLEAR chosen files
22   Select choice once only
24   Maximum 13 programs
26   Reverse filename and place
    choice in K$(K), increment K
28   Check cursor position
30-34 Flash cursor and print
    selection in reverse
36   Menu if F1 pressed
38   Print elements of K$(K), ask
    KILL ABOVE. If yes then
    print wait message, else
    re-run program
40   Read screen into A$(z), menu
    if done

```

```

42   Print out elements of A$(Z),
    w/o A$(0), Poke RUN40 into
    keyboard buffer, kill A$(0),
    then END (negative pokes are
    subtracted from 65536 (used
    to conserve memory)

```

```
44-50 Read screen into array A$(I)
```

```
52-54 Get key routine, if flag is
    set then flash cursor

```

Listing for KILLR.BA.

```

2   'V4 KILLR (C)1986 A. BARBOLLA
4   CLEAR500
    :DIMAS(21)
    :DIMK$(14)
    :KEYON
    :ONKEYGOSUB36
    :RO$=CHR$(27)+"p"
    :RF$=CHR$(27)+"q"
    :W$=RO$+"PLEASE WAIT"+RF$
    :CLS
    :FILES
    :PRINT@294,W$;
    :GOSUB44
    :J$="<KILL>"
    :L$="<CLEAR>"
    :A$(I)=J$
    :A$(I+1)=L$
    :I=I+2
    :CLS
    :FORZ=0TOI
    :PRINT@Z*10,A$(Z)
    :NEXT
    :GOSUB32
6   FG=1
    :GOSUB52
    :ONINSTR(CHR$(29)+CHR$(28)+
    " "+CHR$(30)+CHR$(31)+CHR$(13),
    I$)GOTO8,10,12,14,18
    :GOTO6
8   P=P-10
    :GOTO16
10  P=P+10
    :GOTO16
12  P=P-40
    :GOTO16
14  P=P+40
16  GOSUB28
    :GOTO6
18  IFA$(T)=L$THENRUN
20  IFA$(T)=J$THEN38
22  IFLEFT$(A$(T),1)=CHR$(27)THEN6
24  IFK=13THEN6

```



```

26 EF=1
   :K$(K)=A$(T)
   :A$(T)=RO$+A$(T)+RF$
   :K=K+1
   :GOTO6
28 T=P/10
   :IFT<0ORT>I-1THENP=0
   :T=0
   :GOSUB30
   :GOTO6
30 IFP=LPTHENRETURN
32 PRINT@P,RO$+A$(T);
   :IFFTHENEF=0ELSEPRINT@LP,
     RF$+LA$;
34 LP=P
   :LA$=A$(T)
   :RETURN
36 MENU
38 CLS
   :PRINTRF$
   :FORZ=0TOK
   :PRINT@Z*10,K$(Z)
   :NEXT
   :PRINT@254,"KILL ABOVE?"
   :GOSUB52
   :IFI$="Y"ORI$="y"THENPRINT@254,
     W$ELSERUN
40 CLEAR
   :DIMA$(13)
   :KEYON
   :ONKEYGOSUB36
   :GOSUB44
   :IFI=0THENMENU
42 CLS
   :FORZ=1TOI
   :PRINT@Z*10,A$(Z)
   :NEXT
   :PRINT@80,""
   :E$="RUN40"+CHR$(13)
   :L=LEN(E$)
   :FORP=-85TO-85+(L*2)-1STEP2
   :E=E+1
   :POKEP+1,0
   :POKEP,ASC(MID$(E$,E,1))
   :NEXT
   :POKE-86,L
   :KILLA$(0)
   :END
44 FORS=-512TO-189
   :A$=CHR$(PEEK(S))
   :IFA$="."ORFLTHENFL=FL+1
46 IFA$=" "ORA$="*"THENC=C+1
   :IFC>15THENRETURNELSENEXT

```

```

48 A$(I)=A$(I)+A$
   :IFFL=3THENI=I+1
   :FL=0
   :C=0
50 NEXT
   :RETURN
52 I$=INKEY$
   :IFFGTHENGOSUB32
54 IFI$=""THEN52ELSEFC=0
   :RETURN

```

LEARNING MADE EASY

This educational program from Bill Qualls takes us back to our old schoolroom -- and possibly beyond. -Ed

The need to memorize information is most obvious during our years of schooling. But that same need continues beyond our days in the classroom. STUDY.BA will aid you in any situation in which you need or want to memorize data.

I wrote STUDY.BA to help me to memorize scriptures, but its flexible design lends itself to many uses: children can practice for a spelling test, high school students can learn a foreign language, budding young actors and actresses can practice their lines, choir members can learn the lyrics of hymns, and so on. The uses for STUDY.BA are limited only by your need or desire to learn. Remember, the best way to prepare for a test is to simulate a testing environment.

I am a strong advocate of the "Keep It Simple" approach to program design. Consequently, STUDY.BA is very easy to use. Press the [Esc] key at any time to return to the main menu. The Model 100's text processor is ideally suited for creating and editing

the prompts and text which serve as input to the program. For example, let's assume you want to prepare for your next vacation to Acapulco by brushing up on your spanish. You might enter TEXT and create SPAN.DO as like this:

```
>HELLO
HOLA
>FRIEND
AMIGO
>GOODBYE
ADIOS
```

The greater-than sign in the first position indicates this is a prompt, and that data will be shown. For example, the first thing shown is HELLO. The next line shows a greater-than sign, indicating that a response is expected. You then begin to enter your answer. Your response will be verified one letter at a time. As a correct letter is entered, it is displayed. If you enter the wrong letter, you'll get a BEEP (if beep is turned on -- otherwise it's ignored). If you're stumped, enter a slash (/) and the next letter will be shown. If you want to see the entire answer, enter a question mark (?). Press any key (except Esc) to continue.

If a prompt begins with two greater-than signs (>>) the screen will be cleared before the prompt is displayed. Note there can be more than one prompt. This makes possible a nicely formatted "programmed learning" type of display. For example, a tutorial on statistics may include the information shown in Figure 2. (Note blank lines indicate null strings: simply press [Enter] while in TEXT. The sigma sign [Σ] is produced by pressing [Grph] and [S] simultaneously.)

I use my Model 100 extensively while commuting on the train to Chicago from the suburbs. So as

not to draw undue attention to myself and in consideration of the other commuters I made the BEEP facility of STUDY.BA optional. Upon executing STUDY.BA you will be asked if you want BEEP's. Enter Y or N. You may also find this option useful for dealing with other disinterested parties, such as sleeping spouses and children.

You will then be shown the available files. The .DO extension is optional and will be added if omitted. When you have completed a file, you will again be shown the available files. You can simply press Enter to repeat using the same file, or enter a new text file name, or press Esc to quit.

Please note that STUDY.BA is case sensitive; that is, it will not accept a capital letter in place of a lower case letter. Consequently, for many applications, such as spelling and vocabulary lists, you may want to use the Caps Lock key.

By following these instructions, teachers can store a modified version of STUDY.BA along with a text file on a single cassette tape, thereby creating a fully self-contained instructional unit. You may want to create a backup copy of STUDY.BA prior to making these changes.

- * Delete lines 120, 121, 125 and 126.

- * Delete lines 1000, 1010, 1020, 1030, 1040, 1050 and 1060.

- * Replace line 110 as follows:

```
110 CLS:OPEN"CAS:X.DO"FORINPUTAS1
:GOSUB200:MENU
```

- * Connect cassette recorder.
Press RECORD and PLAY. Enter:
CSAVE"X"

- * Press F8 to return to main menu when save is complete.

* Position cursor over the appropriate text file. Press enter. Save to cassette as follows:

Press F3
Save to: X
Press F8 to return
to main menu
when save is complete.

* To run, go into BASIC and enter:
CLOAD"X",R

Here's a sample script for STUDY.BA -- everything you've ever wanted to know about the mean, median and mode.

Listing of STUDY.BA.

```

1      GOTO 1000
5      PRINT CHR$(95);CHR$(8);
6      X$=INKEY$
      :IF X$="" THEN 6 ELSE IF
      X$=CHR$(27) THEN CLOSE
      :MENU ELSE RETURN
7      IF B=1 THEN BEEP
      :RETURN ELSE RETURN
1000   CLS
      :SCREEN 0,0
      :CLEAR 5000
      :PRINT @41,"STUDY.BA by
      Q-Soft"
      :PRINT @121,"Press [Esc] at
      anytime to Quit"
      :PRINT @201,"Do you want
      BEEP's? (Y/N) ";
105    GOSUB 5
      :IF X$="n" OR X$="N" THEN B=0
      :PRINT X$; ELSE IF X$="y" OR
      X$="Y" THEN B=1
      :PRINT X$; ELSE GOTO 105
110    CLS
      :FILES
      :PRINT " Source file? ";
      :GOSUB 1000
      :ON ERROR GOTO 125

```

```

120    IF INSTR(1,S$,".")=0 THEN
      S$=S$+".DO"
121    OPENS$ FOR INPUT AS 1
      :ON ERROR GOTO 0
      :CLS
      :GOSUB 2000
      :GOTO 110
125    RESUME 126
126    PRINT
      :PRINT "File open ERROR ...Try
      again"
      :GOSUB 7
      :GOSUB 7
      :GOTO 110
2000   IF EOF(1) THEN CLOSE
      :RETURN ELSE LINE INPUT#1,T$
210    IF LEN(T$)=0 THEN GOSUB 6
      :PRINT " "
      :GOTO 2000
220    IF LEN(T$)>1 THEN IF
      LEFT$(T$,2)=">>" THEN CLS
      :T$=RIGHT$(T$,LEN(T$)-1)
230    IF LEFT$(T$,1)=">" THEN PRINT
      RIGHT$(T$,LEN(T$)-1)
      :GOTO 2000
240    H=0
      :PRINT "> ";
      :FOR I=1 TO LEN(T$)
245    IF H=0 THEN GOSUB 5
      :IF X$="?" THEN H=1
250    IF X$=MID$(T$,I,1) OR X$="/"
      OR H=1 THEN PRINT
      MID$(T$,I,1); ELSE GOSUB 7
      :GOTO 245
260    NEXT I
      :GOSUB 6
      :PRINT
      :GOTO 2000
1000    F$=""
1010    GOSUB 5
1020    IF LEN(F$)>0 THEN IF
      X$=CHR$(8) THEN
      F$=LEFT$(F$,LEN(F$)-1)
      :PRINT " ";X$;X$;
      :GOTO 1010
1030    IF X$=>CHR$(32) THEN F$=F$+X$
      :PRINT X$;
1040    IF LEN(F$)<9 AND X$<>CHR$(13)
      THEN 1010
1050    IF F$<>"" THEN S$=F$
1060    RETURN

```

ARE YOU OUT THERE?

When the editorial staff of Portable 100 magazine designed the Portable Program Review Newsletter, we wanted to provide a stream of ready-to-run software and utilities for our readers. PPR is certainly fulfilling that reader need.

We hoped that the offices would be flooded with programs submissions -- useful, well-written and well-documented. Well, that just hasn't happened.

A few months back, I asked for cassette, DVI and disk-drive utilities for a special mass-storage edition of PPR. Not a single submission arrived.

So, I'll repeat that plea. We can only publish what you submit, so please don't be shy about sharing your storage utilities, games, business applications or solutions to problems. There's nothing to lose, and a wealth of software to gain.

Alan L. Zeichick
Editor

PORTABLE PROGRAM REVIEW
P.O. Box 250
Highland Mill
Camden, ME 04843

FIRST CLASS U.S. POSTAGE PAID CAMDEN, ME PERMIT NO. 5

SUBSCRIBER: PLEASE ADVISE IF ABOVE ADDRESS IS INCORRECT

PORTABLE PROGRAM REVIEW is published monthly and copyright © 1986, all rights reserved, by Camden Communications Inc., a Maine corporation with offices at Highland Mill (P.O. Box 250), Camden, Maine 04843. Contents of this publication may not be reproduced in whole or in part unless expressly authorized in writing by the publisher. Tandy and Radio Shack are registered trademarks of Tandy Corporation. U.S. Subscription \$29.97. Please address all subscriptions and inquiries to Nancy Wight, Circulation Director, (207) 236-4365. ISSN: 0883-7295.